



Trabalho Prático 2

Estudo do funcionamento de um processador

ARQUITETURA DE COMPUTADORES

DEPARTAMENTO DE ENGENHARIA ELETRÓNICA E DE
TELECOMUNICAÇÕES E COMPUTADORES

13 de Abril de 2023

1 Objetivos

Este trabalho tem como principal objetivo a compreensão do funcionamento de um processador de um computador, sendo abordados os seguintes tópicos: codificação de um conjunto de instruções, funcionamento de uma microarquitetura e codificação de programas em linguagem máquina.

2 Especificação do Exercício

Pretende-se completar o projeto de um processador, respeitando o seu modelo de programação e microarquitetura. O processador considerado é de 8 bits e tem o seguinte modelo de programação:

- Espaços de endereçamento para código e dados com 256 endereços;
- Oito registos de uso geral, denominados **r0**, **r1**, ... e **r7**;
- Um registo de estado, denominado **CPSR** (do inglês *Current Program Status Register*), que disponibiliza um bit que indica que a última operação realizada produziu resultado zero (*flag Z*);
- O conjunto de instruções apresentado na Tabela 1.

A Figura 1 apresenta o diagrama lógico da microarquitetura do processador, que é de ciclo único.

Instrução	Descrição	
add rd, rn	Adiciona o conteúdo de rn ao conteúdo de rd , colocando o resultado em rd e atualizando o registo CPSR com a informação do bit Z gerada na ULA.	$rd \leftarrow rd + rn$ $rd == 0 ? \text{CPSR.Z} \leftarrow 1 : \text{CPSR.Z} \leftarrow 0$
b rn	Muda a execução para o endereço definido pelo conteúdo de rn .	$PC \leftarrow rn$
bne label	Muda a execução para o endereço associado ao símbolo label quando a <i>flag Z</i> do registo CPSR apresenta o valor zero.	$\text{CPSR.Z} == 0 ? PC \leftarrow label : PC \leftarrow PC + 1$
cmp rn, rm	Subtrai o conteúdo de rm ao conteúdo de rn , atualizando o registo CPSR com a informação do bit Z gerada na ULA.	$rn - rm == 0 ? \text{CPSR.Z} \leftarrow 1 : \text{CPSR.Z} \leftarrow 0$
ldr rd, [rn]	Copia para rd o conteúdo da posição da memória de dados com o endereço definido pelo conteúdo de rn .	$rd \leftarrow M[rn]$
mov rd, #imm3	Estabelece em rd o valor da constante imm3 .	$rd \leftarrow imm3$
push rn	Copia o conteúdo de rn para o topo da pilha, atualizando o valor do registo SP .	$M[-SP] \leftarrow rn$
str rd, [rn]	Copia o conteúdo de rd para a posição da memória de dados com o endereço definido pelo conteúdo de rn .	$M[rn] \leftarrow rd$

Tabela 1: Conjunto de instruções do processador, onde **rd**, **rm** e **rn** representam um dos registos de uso geral do processador, **imm n** simboliza um número natural codificado com n bits, **label** identifica um endereço na vizinhança de ± 32 endereços da instrução de salto em causa, **SP** referencia o registo **r7** e **PC** referencia o registo *program counter*.

3 Trabalho a Realizar

3.1 Análise da microarquitetura

Considere a descrição da microarquitetura do processador apresentada na Figura 1.

1. Comente a seguinte afirmação: "A microarquitetura do processador é do tipo *Harvard*."
2. Indique a funcionalidade do bloco **Ext**, explicitando o propósito do sinal **SE**. Justifique a sua resposta com base no funcionamento das instruções **bne** e **mov**.

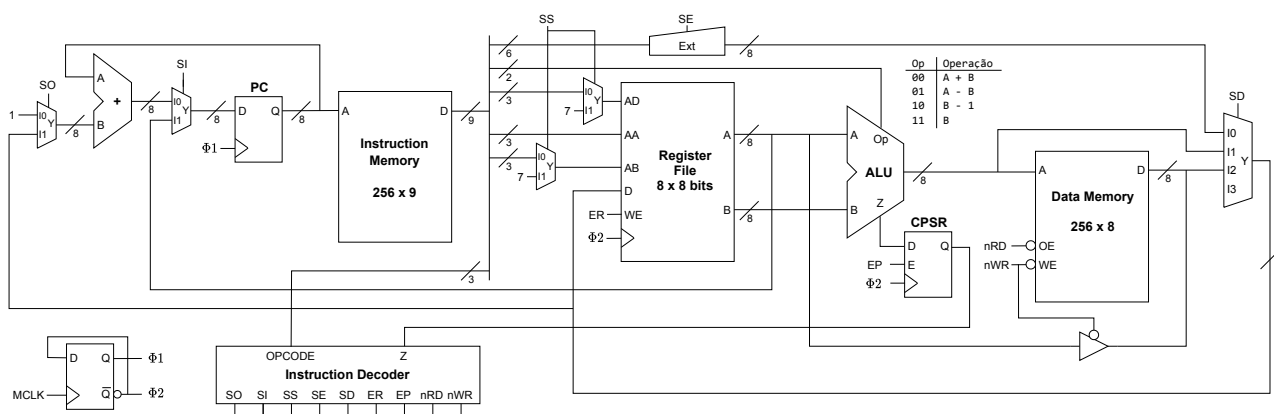


Figura 1: Microarquitetura do processador objeto de estudo.

- Indique o modo de endereçamento associado à instrução **bne label** e discuta as suas vantagens e desvantagens relativamente ao modo de endereçamento associado à instrução **b rn**.

3.2 Codificação das instruções

Considere a utilização de um código de comprimento fixo e um esquema de codificação uniforme para a codificação do conjunto de instruções apresentado na Tabela 1.

- Apresente um mapa de codificação para o conjunto de instruções, tendo em conta o formato de codificação da instrução **add** apresentado na Figura 2.

rd	rn	opcode
8 7 6	5 4 3	2 1 0

Figura 2: Formato de codificação da instrução **add**.

- Sabendo que o valor 100 corresponde ao código de operação (**opcode**) da instrução **add**, indique os valores deste campo para as restantes instruções. Justifique a sua resposta tendo em conta também o funcionamento da Unidade Lógica e Aritmética (ULA).

3.3 Projeto do decodificador de instruções

Considere o subcircuito **Instruction Decoder** apresentado na Figura 1, que implementa o decodificador de instruções da microarquitetura proposta para o processador.

- Usando uma tabela, apresente os valores das saídas deste bloco, em função dos seus sinais de entrada, para o conjunto de instruções apresentado. Explícite os casos de indiferença (*don't care*) e, se aplicável, as saídas obtidas diretamente do código da instrução.
- Considerando uma implementação baseada exclusivamente numa ROM, indique a correspondência entre os bits dos portos de endereço e de dados da memória e os sinais de entrada e de saída do bloco **Instruction Decoder**.
- Indique, em bits, a capacidade da memória ROM considerada no ponto 2.

3.4 Codificação de programas em linguagem máquina

Considere o troço de código na Listagem 1, escrito na linguagem *assembly* do processador.

```
        mov    r0, #0
        mov    r1, #0
        mov    r2, #4
loop:   ldr     r3, [r0]
        add    r1, r3
        mov    r4, #1
        add    r0, r4
        cmp    r0, r2
        bne    loop
        str    r1, [r2]
        mov    r5, #6
        add    r5, r5
        b      r5
```

Listagem 1: Troço de código de teste.

1. Indique a funcionalidade do troço de código.
2. Traduza o troço de código apresentado para código máquina. Use uma tabela com o formato indicado na Tabela 2 para registar o resultado dessa codificação, considerando que cada linha da tabela deverá corresponder apenas a uma instrução do programa e que a instrução `mov r0, #0` deverá ser localizada na posição de memória com o endereço zero. Represente em notação hexadecimal, usando quatro dígitos, os valores dos endereços de memória e das instruções do programa.

Instrução	Endereço	Código máquina
mov r0, #0		

Tabela 2: Tabela exemplo para o registo da codificação para código máquina.

4 Avaliação

O trabalho deve ser realizado em grupo e conta para o processo de avaliação da unidade curricular.

Na entrega do trabalho, cada grupo deverá submeter na plataforma Moodle um relatório do trabalho realizado com as respostas às perguntas formuladas no enunciado, descrevendo, sucintamente, os raciocínios e os cálculos efetuados.

A data limite para a entrega dos trabalhos é 21 de abril de 2023.

Após esta entrega, o docente responsável pela lecionação das aulas teórico-práticas combinará com cada grupo de alunos uma data e hora para a realização da apresentação do trabalho.